Problema E

Torres de Hanoi. Again.

Arquivo fonte: hanoi.{ c | cpp | java | py }

Autor: Prof. Antonio Cesar de Barros Munari (Fatec Sorocaba)

O jogo Torres de Hanoi é bastante conhecido do pessoal que estuda computação. Em geral, é um dos exemplos clássicos utilizados nas aulas sobre recursividade em programação, já que um elegante algoritmo recursivo é capaz de resolvê-lo sem grande dificuldade. Para quem não está associando o nome à pessoa, Torres de Hanoi possui 3 postes fixos, geralmente chamados de A, B e C, e uma certa quantidade finita de discos. Cada disco possui um diâmetro diferente dos demais, e apresenta um furo em seu centro, de maneira que é possível encaixá-lo em qualquer um dos postes que se desejar. Na configuração inicial todos os discos encontram-se encaixados em um dos postes, em ordem de tamanho, com o disco menor em cima e o disco maior embaixo, conforme ilustra a figura 1, onde a configuração dispõe de 5 discos.

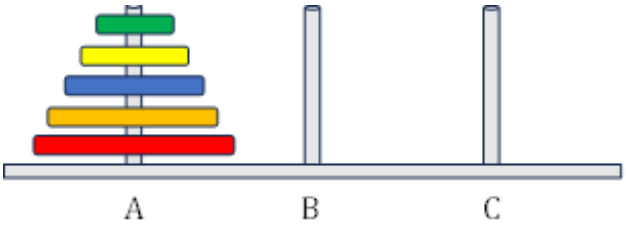

Figura E.1: Estado inicial de uma Torre de Hanoi com 5 discos.

O objetivo do jogo é transferir todos os discos para um poste indicado como destino, sendo permitido mover apenas um disco por vez, utilizando o poste restante como auxiliar para armazenamento e não podendo colocar, em qualquer momento, um disco maior sobre um disco menor no mesmo poste. Existe sempre uma sequência ótima de movimentos para atingir esse objetivo, como mostra a figura 2 onde, para um arranjo de três discos, são necessários sete movimentos. Essa quantidade ótima de movimentos é, como você já deve desconfiar, proporcional à quantidade de discos disponíveis no jogo: para um disco, um movimento; para dois discos, três movimentos; para quatro discos, quinze movimentos e assim por diante. Após cada movimento válido, o jogo assume um novo estado válido, e no processo transitamos de um estado inicial válido para um estado final também válido. É possível descrever cada estado válido do jogo por meio de uma sequência única e específica de zeros e uns, onde o comprimento dessa sequência é igual à quantidade de discos. O primeiro dígito (aquele mais à esquerda) corresponde ao maior disco, o último dígito (aquele mais à direita) representa o menor disco, e os outros dígitos seguem a mesma lógica. Por exemplo, o estado inicial válido para o jogo com 3 discos será 000; se forem 4 discos, será 0000, etc. Considerando o jogo expresso na figura 2, teremos por definição o estado inicial 000; depois do primeiro movimento, teremos 001, que significa que os dois discos maiores estão no mesmo poste e o disco menor está em um outro poste. O segundo movimento levará o jogo para uma configuração 010 (maior disco em um poste, disco intermediário em um poste diferente daquele onde está o maior disco e o disco menor estando em um poste diferente daquele onde está o disco intermediário). O terceiro movimento produzirá o estado 011 (disco maior em um poste, os outros dois discos empilhados em um mesmo outro poste, distinto daquele onde está o maior). O quarto movimento levará a 100 (e aqui perceba um padrão: o disco recém movimentado terá sempre o dígito ‘1’); após o quinto movimento teremos 101; o sexto movimento nos deixará no estado 110 e, finalmente, com o sétimo movimento, encontraremos o estado 111 (todos os discos em um mesmo poste).

Sua tarefa neste problema é bem simples: dada uma sequência de zeros e uns que expressa um estado válido

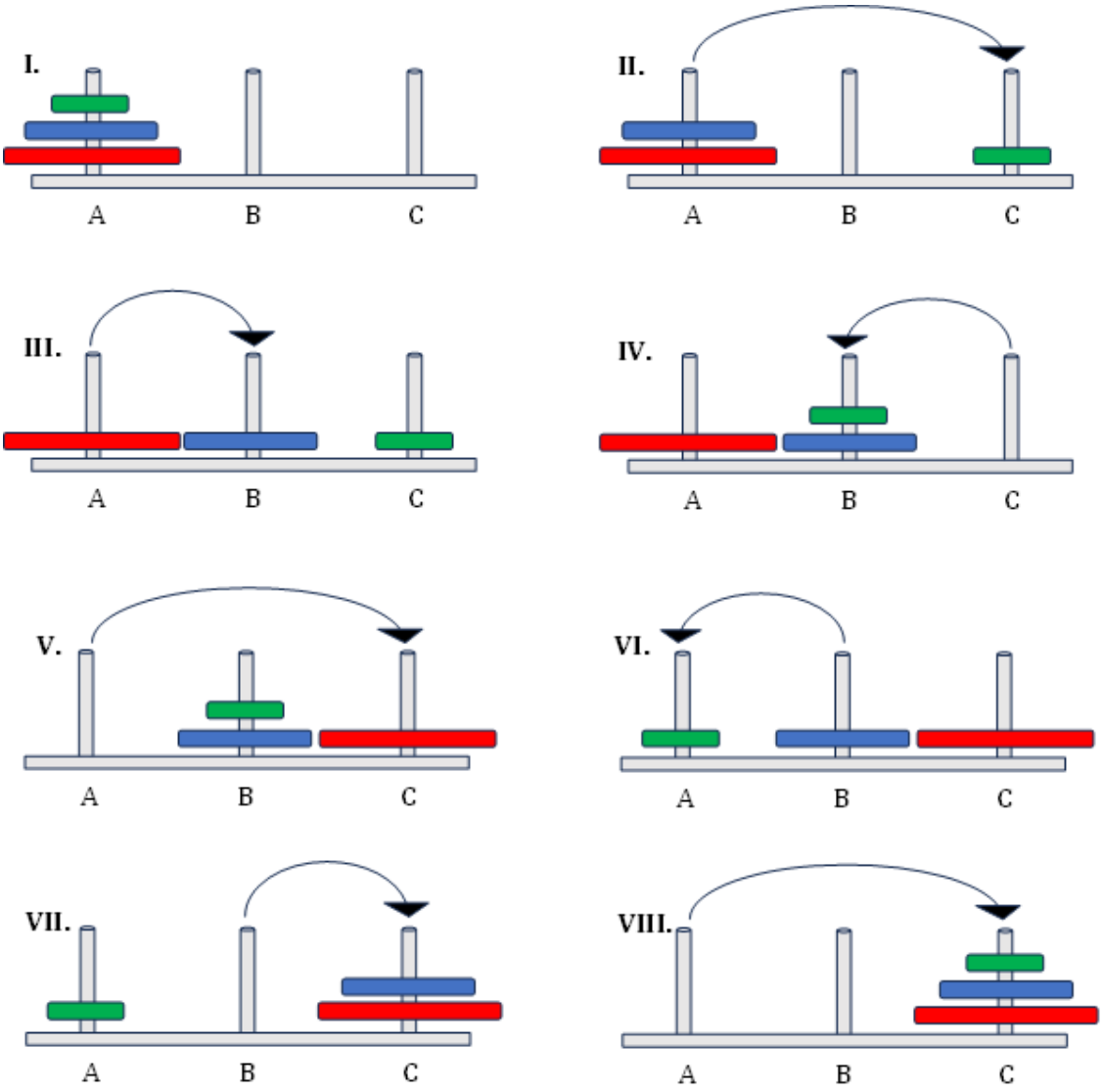

Figura E.2: Solução de uma Torre de Hanoi com 3 discos.

do jogo, indicar quantos movimentos válidos ainda faltam para finalizá-lo, de maneira ótima.

Entrada

Cada arquivo de entrada representa um único caso de teste. Na primeira linha teremos um inteiro N ($0 \leq N \leq 30$) indicando a quantidade de discos. Na segunda linha teremos uma sequência de N dígitos d ($d \in \{0, 1\}$) separados entre si por espaços em branco.

Saída

Imprimir um inteiro X que indica a menor quantidade de movimentos válidos que faltam para o jogo ser finalizado. Não esqueça de deixar uma quebra de linha após o valor impresso.

Exemplo de Entrada 1	Exemplo de Saída 1
3 0 1 0	5

Exemplo de Entrada 2	Exemplo de Saída 2
5 1 0 0 0 1	14

Exemplo de Entrada 3	Exemplo de Saída 3
6 1 0 0 1 1 1	24